

**FIGURE 1**

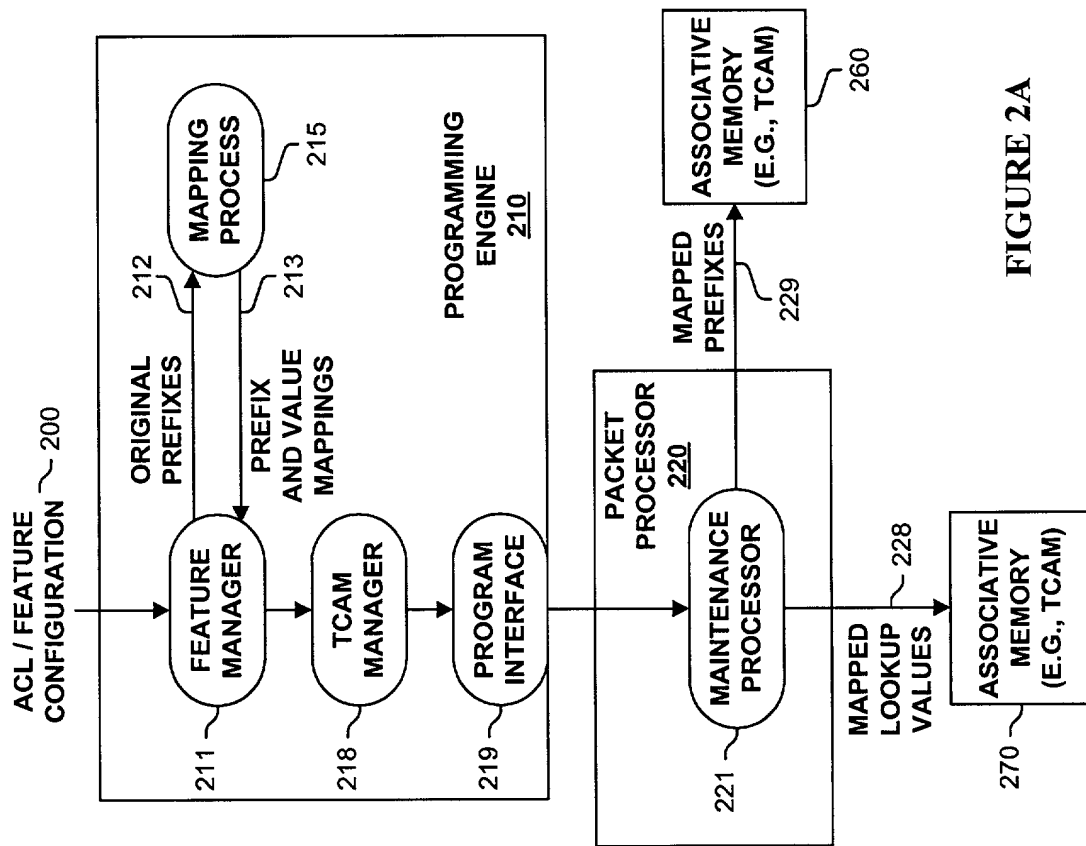


FIGURE 2A

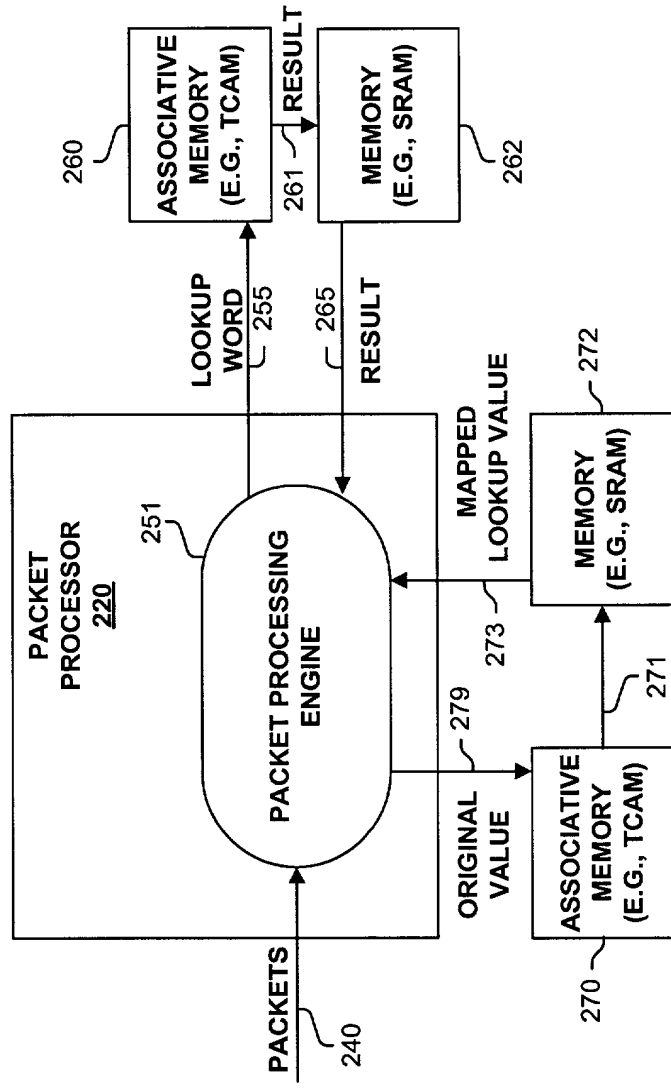
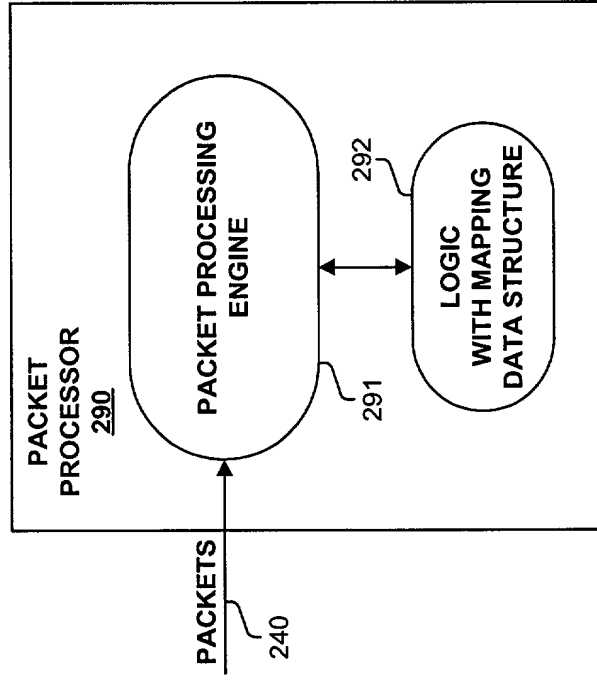


FIGURE 2B



**FIGURE 2C**

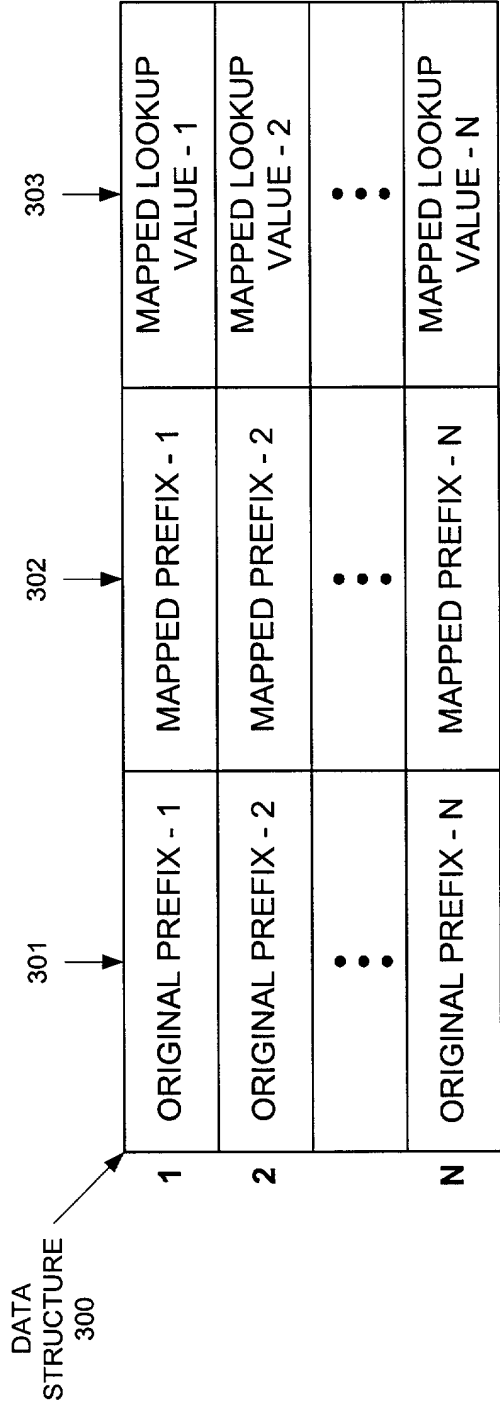
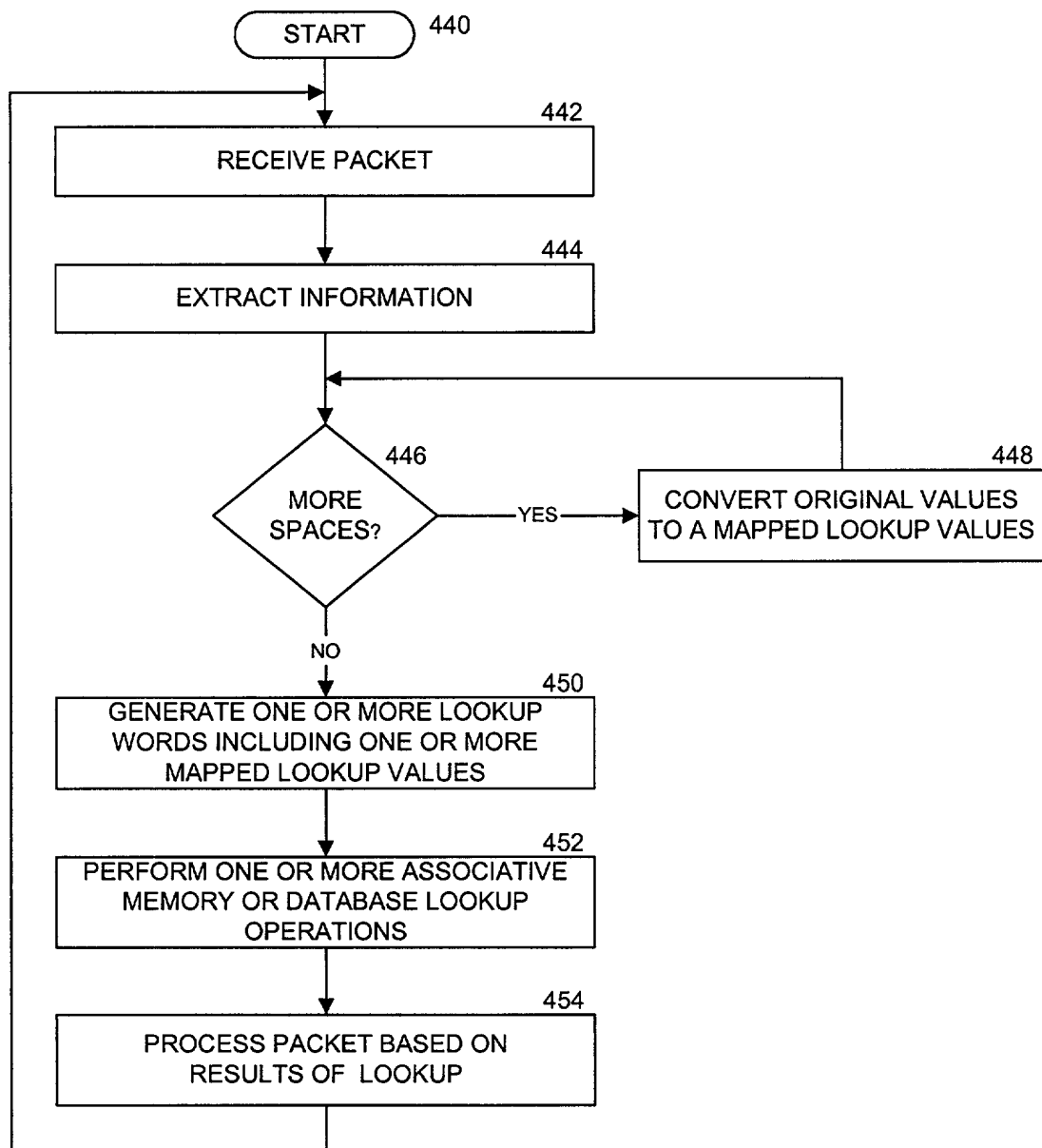
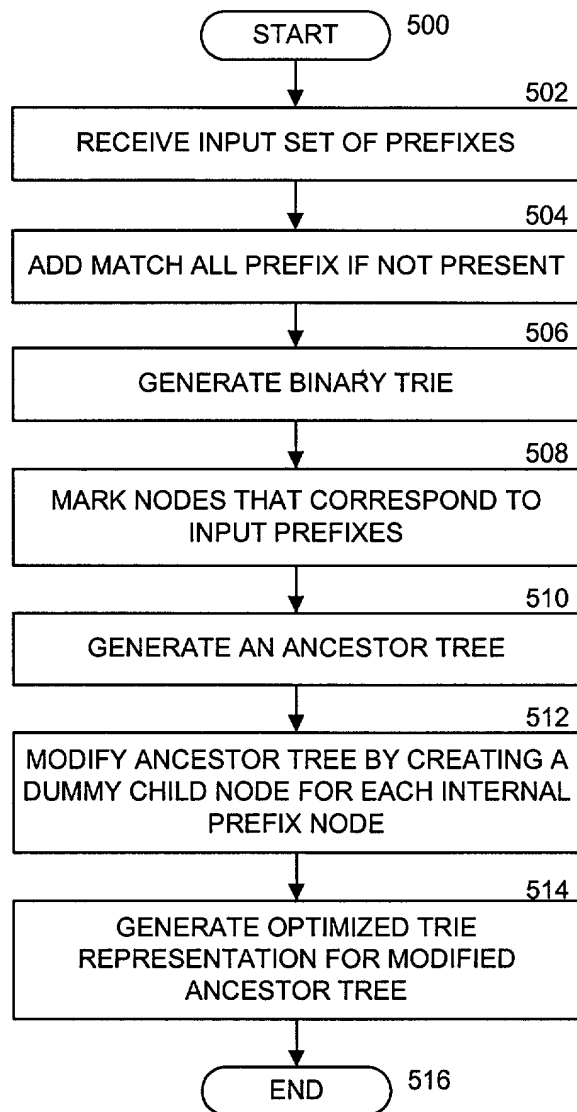


FIGURE 3





**FIGURE 4B**



**FIGURE 5A**



550 →

```

create_anc_tree ( trie_node, anc_tree_node) {
    /* Creates ancestor tree for sub-trie rooted at trie_node, and
     * anchors the ancestor tree at anc_tree_node as root.
     */
    if ( IS_PREFIX(trie_node) )
        anc_tree_node = create_child (anc_tree_root,trie_node);
    if ( NOT_LEAF(trie_node) ) {
        create_anc_tree ( trie_node->left, anc_tree_node);
        create_anc_tree ( trie_node->right, anc_tree_node);
    }
}

```

FIGURE 5B

560 →

```

find_cost(v) {
    list_of_subtrees = { u | u \in children(v) }
    while ( SIZE_OF(list_of_leaves) > 1 ) {
        u = delete_min_cost_element ( list_of_subtrees);
        v = delete_min_cost_element ( list_of_subtrees);
        node = new_node();
        node->left = u; node->right = v;
        node->cost = v->cost + 1;
        insert_element ( list_of_subtrees, node);
    }
    return list_of_elements[0]->cost;
}

```

FIGURE 5C

EXEMPLARY TRIE  
REPRESENTATION OF  
ORIGINAL PREFIXES

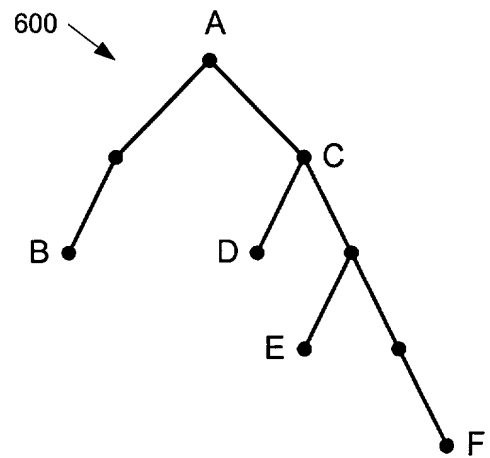


FIGURE 6A

EXEMPLARY  
ANCESTOR TREE

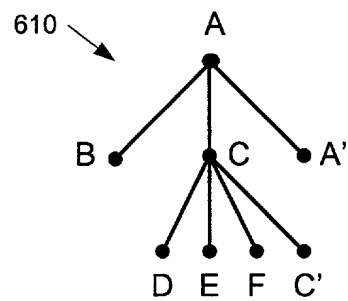


FIGURE 6B

EXEMPLARY  
ANCESTOR TREE COST

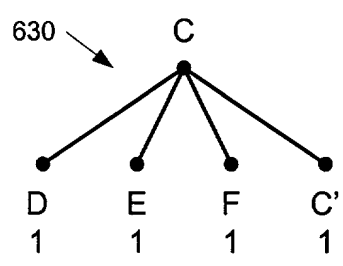


FIGURE 6C

EXEMPLARY OPTIMIZED  
ENCODING OF  
ANCESTOR TREE

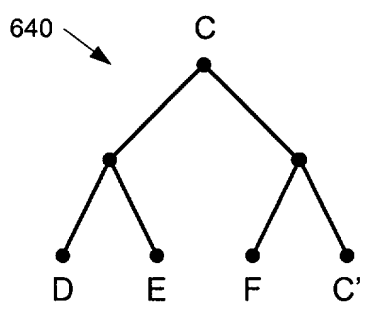


FIGURE 6D

EXEMPLARY TRIE  
REPRESENTATION OF  
MAPPED PREFIXES

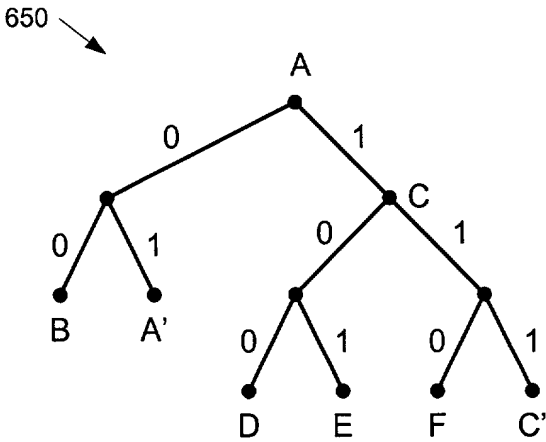


FIGURE 6E

EXEMPLARY NODE IDENTIFIER 661	EXEMPLARY ORIGINAL PREFIX 662	EXEMPLARY MAPPED PREFIX 663	EXEMPLARY LOOKUP VALUE 664
A	*	*	010
B	00	00	000
C	1*	1*	111
D	10	100	100
E	110	101	101
F	1111	110	110

FIGURE 6F